KATJA BERČIČ
MICHAEL KOHLHASE
FLORIAN RABE

TOWARDS A HETEROGENEOUS QUERY LANGUAGE FOR MATHEMATICAL KNOWLEDGE

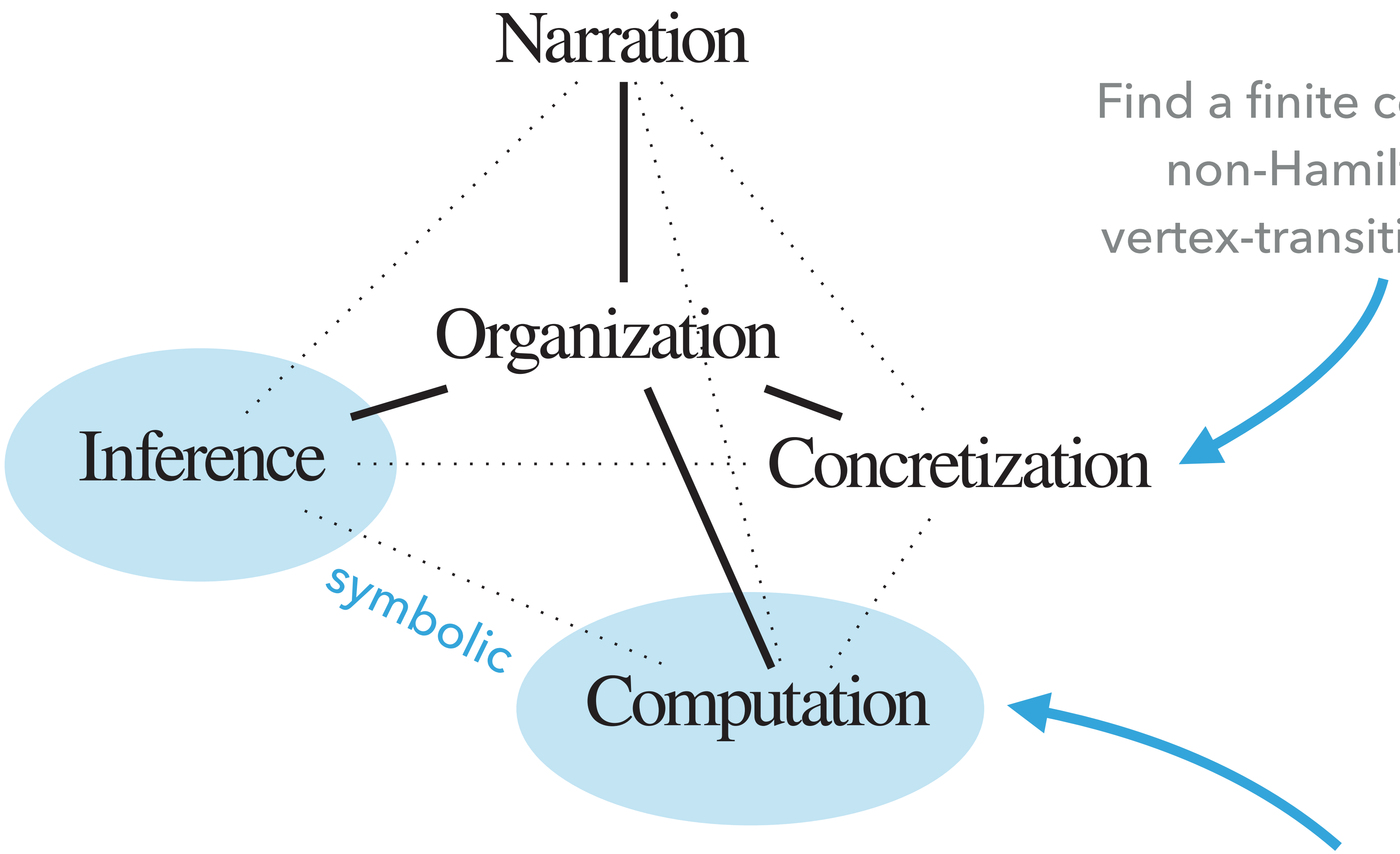
MOTIVATION

- More than 120.000 articles published annually.
- Increasing numbers of active documents and datasets (!)
- *math = cool*, but *math+computers = better*

HOWEVER:

- Existing search systems focus on only one aspect.
- Often more is needed: querying heterogeneous mathematical knowledge

THE TETRAPOD OF DOING MATHEMATICS



Find a finite connected
non-Hamiltonian
vertex-transitive graph

What has the property
$F_n = F_{n-1} + F_{n-2}$

OUR CONTRIBUTION

- A tractable design of a query language for mathematics with a corresponding architecture that spans over all kinds of knowledge
- Subsumes formula search (like MathWebSearch) or even formula search combined with bag of words search
- Less than solving general querying over combined relational databases and triple stores

OEIS

%I A000045 M0692 N0256

%S A000045 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987

%N A000045 Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1.

%C Also sometimes called Lamé's sequence.

%D A000045 V. E. Hoggatt, Jr., Fibonacci and Lucas Numbers. Houghton, Boston, MA, 1969.

%F A000045 F(n) = ((1+sqrt(5))^n-(1-sqrt(5))^n)/(2^n*sqrt(5))

%F A000045 G.f.:Sum{n>=0}x^n*Product{k=1..n}(k+x)/(1+k*x).- Paul D. Hanna, Oct 26 2013

%F A000045 This is a divisibility sequence; that is, if n divides m, then a(n) divides a(m)

%A A000045 N. J. A. Sloane, Apr 30 1991

ONE ASPECT: FIND MATHEMATICAL STRUCTURES

- Table of graphs containing
  - graph encoded as sparse6
  - common (human readable) names of the graph
  - some graph invariants, including arc-transitivity (a boolean)
- Query: find *arc-transitive* graphs

simple SQL!

ONE ASPECT QUERIES: PAPER SEARCH

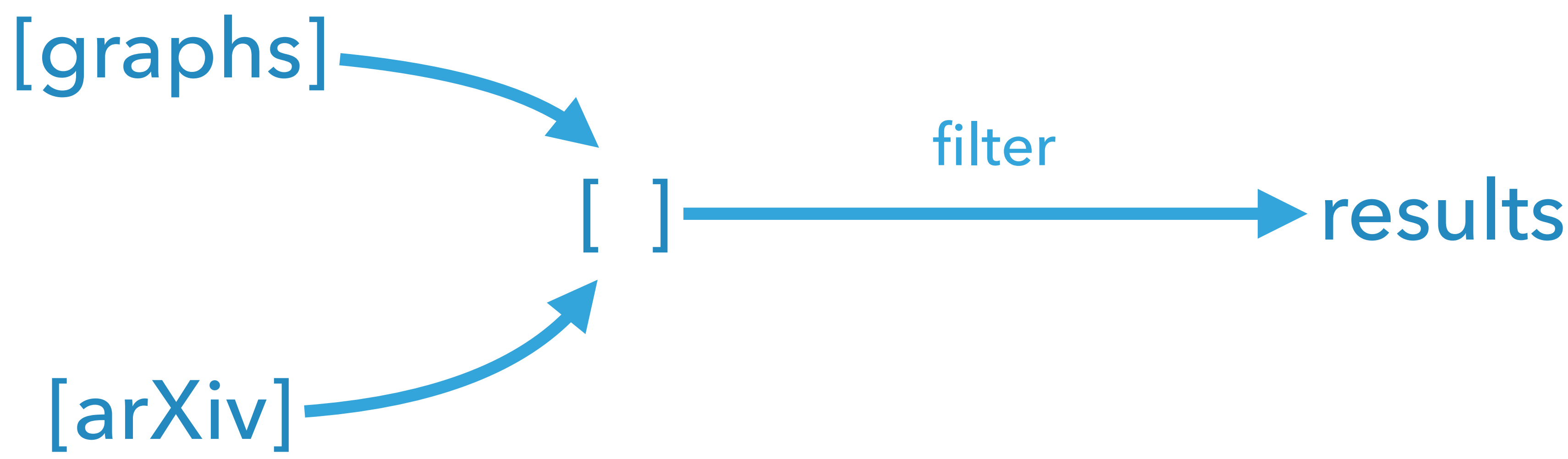
- arXiv index containing
  - narrative index for text
  - formula index (MathWebSearch)
  - metadata knowledge graph
- Query: find papers about the *Petersen graph*
  [with h-index greater than 50]

keyword

metadata from the knowledge graph

MULTIPLE ASPECTS

- Query: find *arc-transitive* graphs mentioned by *name* in articles with *h-index greater than 50*



- Query: find *recent* theorems about *integer sequences* that *contain prime numbers and* satisfy the formula $F_n = F_{n-1} + F_{n-2}$

TERMINOLOGY

- **Document**: file or similar resource containing information; can have comments, metadata.

formalization, theory
source files,
database, ABox,
document, website

- **Library**: (usually) structured collection of documents, grouped by user access

- **Fragment**: part of a document, its internal structure allows defining occurrences of objects

theorem, definition
class, function
table, row, cell
section, paragraph

INDEXING INFORMATION

- **Indexer**: data structure O for indexable objects and a function mapping libraries to sets of index entries.
- **Index entry**: object in a fragment, fragment URI, information about fragment location
- **Index**: set of all entries
- **Query**: object $\Gamma \vdash q : O$, where Γ are the variables
- Result: index entry with object o, together with a substitution for Γ such that q matches o

ORGANIZATIONAL

- Information: organisational metadata and cross-refs
- Stored in: GraphDB, any triple store
- Atomic queries: triples *subject, predicate, object*, possibly containing query variables Q_i
- Examples of atoms:
  Q is a query variable representing a paper
  - "Petersen graph" partOf Q
  - Q bibo: publishedIn "Electronic Journal of Combinatorics"

NARRATIVE

- Information: n-grams of words shingles?
- Stored in: text indexes, eg. Elasticsearch
- Atomic queries: $F \in BagOfWords(W_1, \ldots W_n)$, where F is a query variable representing the fragment in the result set for the bag of words

note: no variables!

- Example of an atom:
  F ∈ BagOfWords("Petersen", "graph") finds all fragments F in which the words "Petersen" and "graph"

SYMBOLIC

- Information: symbolic expressions, formulas, proofs
- Stored in: substitution tree, eg. MathWebSearch
- Atomic queries: $F \in S(Q_1, \ldots, Q_n)$, where S is an expression, Q_i are substitution variables, and F is a query variable representing the fragment in which a unifying expression occurs
- Example of an atom:

  $F \in \sum_i Q \frac{x^i}{i!}$ finds all fragments F containing exponential generating functions with arbitrary coefficients Q

CONCRETE

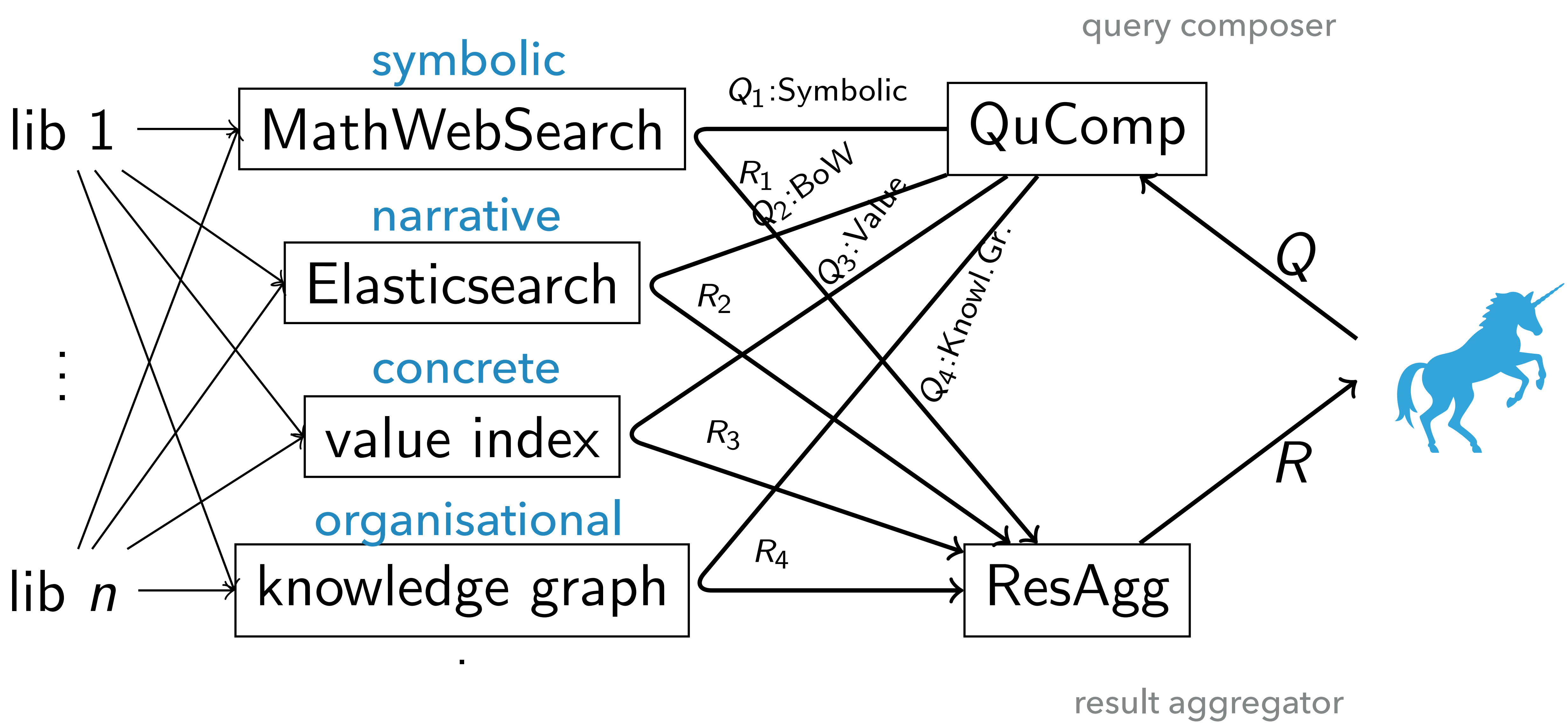
- Information: concrete objects, eg. numbers polynomials, groups, graphs
- Stored in: currently no universal indexing solution, ad hoc indexes for each database
- Example of questions one might ask: find *arc-transitive* graphs

CONCRETE

- But: MathDataHub system aiming at a universal index for all kinds of datasets.
- Idea: for any type, store objects of that type, together with some precomputed properties and information in which datasets they appear
- Atomic queries: SQL-like
- Example: SELECT Q: Graph WHERE arcTransitive(Q)

ARCHITECTURE

every library indexed in every aspect



MULTI-ASPECT QUERY WITH COMMON VARIABLES

```
SELECT G : Graph
WHERE
   arcTransitive(G),
   F ∈ Narr(Name(G), ”graph”),
   F partOf P,
   P bibo: publishedIn J,
   J spar: hasHindex H,
   H > 50
```

find *arc-transitive* graphs mentioned by *name* in articles with *h-index* > *50*

POSSIBLE CONVERSIONS

	organisational	symbolic	concrete
org.	as is	ids, literals: as is other: evaluate	
symbolic	as is	as is	decode
concrete	literals: as codes ids: fail	encode (partial)	as is
narrative	ids: name as string literals: as string other symbolic: evaluate		value as string

OPEN PROBLEMS IN INDEXING CONCRETE VALUES

- Special indexing techniques probably required for certain types and operations (subsequences in OEIS)
- Possible to choose a standard codec for every type, but this will not always be efficient (sparse vs. dense graphs and polynomials, ...)
- Exact vs. approximate values: $e > 2$?

CONCLUSIONS

- Mathematical information retrieval needs to address multiple aspects
- Libraries are typically focused on one aspect, but contain material of other aspects
- The language allows for sharing variables between the aspect-specific sub-queries
- Next step: an implementation of a distributed cross-aspect search engine (as sketched) as part of the MathHub system